

Efficient Time Domain Decomposition Algorithms for Parabolic PDE-Constrained Optimization Problems [☆]

Jun Liu^{a,*}, Zhu Wang^b

^a*Department of Mathematics and Statistical Sciences, Jackson State University, Jackson, MS 39217, USA.*

^b*Department of Mathematics, University of South Carolina, Columbia, SC 29208, USA.*

Abstract

Optimization with time-dependent partial differential equations (PDEs) as constraints appear in many engineering applications. The associated first-order necessary optimality system consists of one forward and one backward time-dependent PDE coupled with optimality conditions. An optimization process by using the one-shot method determines the optimal control, state and adjoint state at once, while, with the cost of solving a large scale, fully discrete optimality system. Hence, such one-shot method could easily become prohibitive when the time span is long or a small time step is taken. To overcome this difficulty, we propose in this paper several time domain decomposition algorithms for improving its computational efficiency. In these algorithms, the optimality system is split into many small subsystems over a much smaller time interval, which are coupled by appropriate continuity matching conditions. Both one-level and two-level multiplicative and additive Schwarz algorithms are developed for iteratively solving the decomposed subsystems in parallel. In particular, the convergence of the one-level multiplicative and additive Schwarz algorithms without overlap are proved. The effectiveness of our proposed algorithms is demonstrated by both 1D and 2D numerical experiments, where the developed two-level algorithms show very scalable convergence rates with respect to the number of subdomains.

Keywords: PDE-constrained optimization, forward-and-backward PDE system, time domain decomposition algorithm, Schwarz algorithms, preconditioners, coarse space correction

1. Introduction

Many scientific and engineering applications involve the optimization of systems governed by time-dependent partial differential equations (PDEs) [1], such as the control and optimization of flows [2], the optimal heating and cooling of minimizing the energy consumption subject to the achievement of a prescribed temperature distribution, and the airfoil design of minimizing the drag subject to a minimal lift. One way of treating such constrained optimization problems is to construct an unconstrained optimization problem through the Lagrange multiplier method [3]. By employing the optimize-then-discretize approach, the corresponding first-order necessary optimality system is often given by a forward-and-backward in time PDE system. In such a system, the state equations are posed forward with initial conditions, the adjoint

[☆]This research was started at the Institute for Mathematics and its Applications (IMA) when J. L. and Z. W. took the IMA new directions short course – topics in control theory in May 2014. We thank IMA for the support. The IMA receives funding from the NSF under Award DMS-0931945.

*Corresponding author

Email addresses: jun.liu@jsums.edu (Jun Liu), wangzhu@math.sc.edu (Zhu Wang)

equations are posed backward with terminal conditions at the final time, and they are coupled by optimality conditions.

The optimal state and control variables can be directly determined from the discrete optimality system. This approach is referred to as the one-shot method. However, since the system has twice as many unknowns as the state equation and is fully discrete in both space and time, the one-shot method needs to solve a large scale system. Therefore, sensitivity- and adjoint-based iterative optimization methods have been developed [2], which solve the state equations and adjoint state equations alternatively so that the computational burden at each optimization iteration is affordable. Nevertheless, the one-shot method is still very appealing, in particular, for convex problems, because a single solving process would be sufficient to obtain the optimal states, adjoint states, and optimal control all together, which doesn't require any inherently sequential optimization iterations. In order to improve its overall efficiency in solving large-scale PDE-constrained optimization, many efficient direct/iterative linear/nonlinear solvers have been proposed, see for example [4, 5, 6, 7, 8, 9, 10, 11] and the reference therein. It was shown in [9] that, when the multigrid method is used, coarsening in space only (semi-coarsening) yields better convergence than coarsening in both space and time. Thus, it is quite natural to investigate numerical remedies, such as time parallel computing, that could relieve the high computational cost due to the time discretization.

Parallel-in-time methods have been investigated for evolution problems over the last four decades. Their central idea is to distribute a tremendous computational task into many small parts, which can be executed by multiple processors simultaneously. Since Nievergelt proposed in [12] the first time decomposition algorithm for finding the parallel solutions of evolution differential equations, the methods of time-parallel time integration have been extensively expanded. The reader is referred to [13] for a survey of these methods. For the forward-backward PDEs, a time parallel iterative method was studied in [14], which is based on a multiple shooting reformulation of the linear quadratic optimal control problems. A domain decomposition in time algorithm was investigated in [15], which follows the discretize-then-optimize approach and employs the additive Schwarz preconditioner in time based on the discrete system. More recently, some preliminary convergence analysis on Schwarz methods in time for linear parabolic control problems was conducted in [16, 17], where an appealing mesh-independent convergence rate is proved under some restrictive technical assumptions on the discretized spatial operators. We note that domain decomposition algorithms in space has been well established ([18, 19, 20, 21, 22, 23]) in solving large systems. But time domain decomposition methods have been less thoroughly investigated in the context of forward-and-backward optimality system.

Therefore, we put forth a time domain decomposition framework for the forward-and-backward PDEs appeared in optimality systems. The proposed approach roots in an equivalent hybrid formulation of the continuous system. Four different types of iterative domain decomposition algorithms are designed to compute numerical solutions by multiple processors in parallel: multiplicative and additive Schwarz algorithms with or without overlap. Since the one-level domain decomposition algorithms would require more iterations for convergence as the number of subdomains increases, to obtain better performance with high degree of parallelism, it is often necessary to use two-level domain decomposition algorithms equipped with a coarse grid correction procedure [18, 24, 19, 11, 25]. Therefore, both one-level and two-level algorithms are numerically investigated in this paper. In particular, two-level domain decomposition algorithms based on some recently developed coarse space correction techniques are implemented [26, 27]. Although it has been pointed out in [15, 17] that two-level algorithms are needed to achieve better scalable convergence rates than one-level algorithms, there are very few work discussing such two-level time domain decomposition algorithms in the setting of forward-and-backward optimality system. For instance, some two-level space-time parallel do-

main decomposition preconditioners are proposed in [25, 11] very recently, where the full discretizations are based on first-order accurate backward Euler scheme in time and the convergence analysis of the algorithms were not given. Different from the above mentioned methods, our proposed time domain decomposition algorithms are based on a second-order accurate leapfrog scheme in time and are proved to be convergent as stand-alone solvers. Moreover, the construction of our coarse grid correction is very different from the geometrical ones used in [8, 25, 11, 28]. The reported comprehensive numerical comparison of four types of domain decomposition algorithms (both one-level and two-level) as stand-alone solvers and preconditioners is another major contribution of this work. In particular, compared to overlapping methods, two-level non-overlapping domain decomposition methods [29, 30] have been rarely discussed in the literature. Our current work contributes to extend the applicability of non-overlapping methods.

The rest of the paper is organized as follows. In Section 2, we consider a typical linear parabolic PDE constrained optimization problem and present its forward-and-backward optimality PDE system. In Section 3, a hybrid formulation with two subdomains is proved to motivate our time domain decomposition algorithms. Based on the developed hybrid formulation, four different domain decomposition algorithms are proposed and some convergence analysis are given in Section 4; two-level algorithms are studied in Section 5. In Section 6, some comprehensive numerical experiments are conducted to demonstrate the effectiveness of our proposed domain decomposition algorithms as stand-alone solvers and preconditioners, respectively. Finally, some concluding remarks are given in the last section.

2. Parabolic PDE-Constrained Optimization and its Discretization

We exemplify our proposed algorithms through discussing a typical distributed control problem. Let Ω be a bounded subset of \mathbb{R}^d ($1 \leq d \leq 3$) with Lipschitz boundary $\Gamma := \partial\Omega$. Define $Q = \Omega \times (0, T)$ and $\Sigma = \Gamma \times (0, T)$ for a given finite period of time T . We consider the following unconstrained optimal control problem [31, 32] for minimizing a tracking-type quadratic cost functional

$$J(y, u) = \frac{1}{2} \|y - g\|_{L^2(Q)}^2 + \frac{\gamma}{2} \|u\|_{L^2(Q)}^2 \quad (1)$$

subject to a linear parabolic PDE system

$$\begin{cases} -\partial_t y + \Delta y = f + u & \text{in } Q, \\ y = 0 & \text{on } \Sigma, \\ y(\cdot, 0) = y_0 & \text{in } \Omega, \end{cases} \quad (2)$$

where $g \in L^2(Q)$ is the desired tracking trajectory, $u \in U := L^2(Q)$ is the distributed control function, $\gamma > 0$ represents either the weight on the cost of control or the Tikhonov regularization parameter, $f \in L^2(Q)$ is the source term, and the initial condition $y_0 \in H_0^1(\Omega)$. For simplicity, we also assume no constraints on the control u . The existence and uniqueness of the solution to the above problem is well established [31, 32].

By defining a Lagrange functional and making use of the strict convexity of the objective functional J , the optimal solution pair (y, u) to (1)-(2) can be uniquely determined by the first-order necessary condition.

It leads to the following PDE system:

$$\begin{cases} -\partial_t y + \Delta y - p/\gamma = f & \text{in } Q, \\ y = 0 & \text{on } \Sigma, \quad y(\cdot, 0) = y_0 \text{ in } \Omega, \\ \partial_t p + \Delta p + y = g & \text{in } Q, \\ p = 0 & \text{on } \Sigma, \quad p(\cdot, T) = 0 \text{ in } \Omega, \end{cases} \quad (3)$$

where the state y evolves forward in time and adjoint state p marches backward. The optimal control u can be derived from the optimality condition $\gamma u = p$ in Q . The fact that the state y and the adjoint state p are marching in opposite orientations represents the main challenge for solving (3) numerically, since we have to resolve all time steps simultaneously in computation, which inevitably results in a huge sparse system of linear equations upon full discretization.

Following [33, 34], we use a second-order leapfrog central finite difference scheme to obtain a full discretization of the optimality system (3). Here, we briefly describe the scheme for completeness and refer interested readers to [33] for more details. The time interval $[0, T]$ is uniformly divided into N sub-intervals by the points $0 = t_0 < t_1 < \dots < t_N = T$ and time step size $\tau = T/N$. The space domain $\Omega = [0, 1]^2$ is partitioned uniformly by grid points (ξ_i, ζ_j) with $0 \leq i, j \leq M$, where $\xi_i = ih$, $\zeta_j = jh$, and $h = 1/M$. Denote the partitioned space domain by Ω_h . For any grid function $Z(\cdot, \cdot)$ on Ω_h , we define the discrete gradient (∇_h) by the forward difference and define the discrete Laplacian (Δ_h) by a 5-point stencil central difference. The full discretization yields

$$-\frac{Y^{n+1} - Y^{n-1}}{2\tau} + \Delta_h Y^n - P^n/\gamma = F^n, \quad n = 1, 2, \dots, N-2, \quad (4)$$

$$\frac{P^{n+1} - P^{n-1}}{2\tau} + \Delta_h P^n + Y^n = G^n, \quad n = 2, 3, \dots, N-1, \quad (5)$$

where $(Y^n)_{ij}$ and $(P^n)_{ij}$ are the discrete approximation of $y(\xi_i, \zeta_j, t_n)$ and $p(\xi_i, \zeta_j, t_n)$, respectively, for $i, j = 1, \dots, M-1$. Similar notations are used for F^n and G^n . The Dirichlet boundary conditions are imposed at the grid points with $i = 1, M-1$ or $j = 1, M-1$ through the discrete Laplacian terms. Also notice that Y^0 and P^N are obtained directly from the given initial conditions $y(\cdot, 0) = y_0$ and $p(\cdot, T) = 0$. To close the above linear system, we impose two additional equations derived from approximating (3) at the last time step (aligned with $n = N-1$ in (4) and $n = 1$ in (5)) by a second-order, one-sided BDF2 scheme [35], i.e.,

$$-\frac{-Y^{N-3} + 4Y^{N-2} - 3Y^{N-1}}{2\tau} + \Delta_h Y^{N-1} - P^{N-1}/\gamma = F^{N-1}, \quad (6)$$

$$\frac{3P^1 - 4P^2 + P^3}{2\tau} + \Delta_h P^1 + Y^1 = G^1. \quad (7)$$

For the purpose of obtaining better performance in our domain decomposition algorithms, we have slightly modified the original scheme in [33] by not treating Y^N and P^0 as unknowns anymore. Otherwise, according to our numerical tests, the given initial conditions $y(\cdot, 0) = y_0$ and $p(\cdot, T) = 0$ can not be easily shifted to the right-hand-side, which seems to introduce large approximation errors on the subdomain interfaces and hence cause deteriorated convergence rates for the 2-level algorithms based on coarse grid residual correction.

The above scheme (4-7) in a matrix form is a two-by-two block, structured, sparse linear system, whose

coefficient matrix is non-symmetric and indefinite, as follows.

$$L_h w_h := \begin{bmatrix} A_h & -I_h/\gamma \\ I_h & D_h \end{bmatrix} \begin{bmatrix} y_h \\ p_h \end{bmatrix} = \begin{bmatrix} f_h \\ g_h \end{bmatrix} =: b_h, \quad (8)$$

where

$$A_h = \begin{bmatrix} \Delta_h & -I/2\tau & \cdots & 0 & 0 & 0 \\ I/2\tau & \Delta_h & -I/2\tau & \cdots & 0 & 0 \\ 0 & I/2\tau & \Delta_h & -I/2\tau & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & I/2\tau & \Delta_h & -I/2\tau \\ 0 & 0 & \cdots & -I/2\tau & 4I/2\tau & (\Delta_h - 3I/2\tau) \end{bmatrix},$$

$$D_h = \begin{bmatrix} (\Delta_h - 3I/2\tau) & 4I/2\tau & -I/2\tau & 0 & \cdots & 0 \\ -I/2\tau & \Delta_h & I/2\tau & 0 & \cdots & 0 \\ 0 & -I/2\tau & \Delta_h & I/2\tau & \ddots & \vdots \\ 0 & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \ddots & -I/2\tau & \Delta_h & I/2\tau \\ 0 & 0 & \cdots & 0 & -I/2\tau & \Delta_h \end{bmatrix},$$

$$f_h = \begin{bmatrix} f^1 - y_{0,h}/2\tau \\ f^2 \\ \vdots \\ f^{N-1} \end{bmatrix}, g_h = \begin{bmatrix} g^1 \\ g^2 \\ \vdots \\ g^{N-1} \end{bmatrix}, y_h = \begin{bmatrix} y^1 \\ y^2 \\ \vdots \\ y^{N-1} \end{bmatrix}, \quad \text{and} \quad p_h = \begin{bmatrix} p^1 \\ p^2 \\ \vdots \\ p^{N-1} \end{bmatrix}.$$

Here I and I_h are identity matrices of appropriate size and the vectors $y_{0,h}$, f^n , g^n , y^n and p^n are the lexicographic ordering (vectorization) of the corresponding function approximations on spatial grid points (in matrix form) at time step t_n . Notice that the initial conditions $y(\cdot, 0) = y_0$ and $p(\cdot, T) = 0$ are already shifted to the right-hand-side, i.e., the first term of f_h and the last term of g_h . It is not difficult to see that the size of the above system (8) will become prohibitively large when N or T gets large enough, which motives our proposed time domain decomposition algorithms in this paper. In some sense, the proposed domain decomposition algorithms in time complements very well with the developed semi-coarsening multigrid algorithm in space [33], as shown in subsection 6.2.

3. A Hybrid Continuous Formulation of Optimality System with Two Subdomains

In this section, we assume the time interval $[0, T]$ is decomposed into two subdomains, $R_1 = (0, T_1^r)$ and $R_2 = (T_2^l, T)$, where $T_1^r \geq T_2^l$. When $T_1^r = T_2^l$, there is no overlapping between two subdomains; when $T_1^r > T_2^l$, there is an overlapping region. Let y_i , p_i be the local solutions on each subdomain $Q_i = \Omega \times R_i$, for $i = 1, 2$. We introduce a hybrid formulation that is the restriction of the optimality system (3) on each subdomain Q_i together with the continuity matching conditions on the interfaces or regions of overlap between adjacent subdomains. We will first show that the hybrid formulation is equivalent to the original system, i.e., the local solutions of the hybrid problem coincide the global solution restricted on each

subdomain. We then show that time domain decomposition algorithms based on the hybrid formulation leads to a solution convergent to the solution of original system, which is not surprising but very fundamental to justify and analyze the convergence of any domain decomposition algorithms.

Theorem 3.1. *Let $y(x, t), p(x, t) \in W(0, T; Q) := \{y \in H^{1,0}(Q), \partial_t y \in L^2(0, T; H^1(\Omega)^*)\}$ be the solutions to optimality system (3). Suppose $y_1(x, t), p_1(x, t)$ and $y_2(x, t), p_2(x, t)$ solve the following hybrid systems*

$$\begin{cases} -\partial_t y_1 + \Delta y_1 - p_1/\gamma = f & \text{in } Q_1, \\ y_1 = 0 & \text{on } \Sigma, \quad y_1(\cdot, 0) = y_0 \text{ in } \Omega, \\ \partial_t p_1 + \Delta p_1 + y_1 = g & \text{in } Q_1, \\ p_1 = 0 & \text{on } \Sigma, \quad p_1(\cdot, T_1^r) = p_2(\cdot, T_1^r) \text{ in } \Omega, \end{cases} \quad (9)$$

and

$$\begin{cases} -\partial_t y_2 + \Delta y_2 - p_2/\gamma = f & \text{in } Q_2, \\ y_2 = 0 & \text{on } \Sigma, \quad y_2(\cdot, T_2^l) = y_1(\cdot, T_2^l) \text{ in } \Omega, \\ \partial_t p_2 + \Delta p_2 + y_2 = g & \text{in } Q_2, \\ p_2 = 0 & \text{on } \Sigma, \quad p_2(\cdot, T) = 0 \text{ in } \Omega, \end{cases} \quad (10)$$

which are coupled through the continuity matching conditions. Then, there hold

$$\begin{cases} y_1 = y \text{ and } p_1 = p & \text{in } Q_1, \\ y_2 = y \text{ and } p_2 = p & \text{in } Q_2. \end{cases} \quad (11)$$

Proof. If y and p are solutions of the optimality system (3), $y_i = y$ and $p_i = p$ on Q_i , then y_1, y_2 and p_1, p_2 will obviously satisfy the hybrid formulation (9)-(10) by the construction.

To prove the converse, we suppose that y_1, y_2 and p_1, p_2 solve the hybrid formulation (9)-(10) and will show that y and p satisfy (3) if $y = y_i$ and $p = p_i$ on Q_i . In particular, we need to show that $(y, p) \in W(0, T; Q)$ satisfies the weak formulation of the optimality system (3):

$$\begin{aligned} \iint_Q y v_t - \iint_Q \nabla y \cdot \nabla v - \frac{1}{\gamma} \iint_Q p v &= \iint_Q f v + \int_{\Omega} y(\cdot, T) v(\cdot, T) \\ &\quad - \int_{\Omega} y(\cdot, 0) v(\cdot, 0), \quad \forall v \in W_2^{1,1}(Q). \end{aligned} \quad (12)$$

$$\begin{aligned} - \iint_Q p w_t - \iint_Q \nabla p \cdot \nabla w + \iint_Q y w &= \iint_Q g w + \int_{\Omega} p(\cdot, 0) w(\cdot, 0) \\ &\quad - \int_{\Omega} p(\cdot, T) w(\cdot, T), \quad \forall w \in W_2^{1,1}(Q). \end{aligned} \quad (13)$$

(I) We first consider the non-overlapping decomposition in time domain, i.e., $T_1^r = T_2^l$. By testing the equations of $y_1 \in W(0, T_1^r; Q_1)$ and $y_2 \in W(T_2^l, T; Q_2)$, (9) and (10), by v respectively, we obtain

$$\begin{aligned} \iint_{Q_1} y_1 v_t - \iint_{Q_1} \nabla y_1 \cdot \nabla v - \frac{1}{\gamma} \iint_{Q_1} p_1 v &= \iint_{Q_1} f v + \int_{\Omega} y_1(\cdot, T_1^r) v(\cdot, T_1^r) \\ &\quad - \int_{\Omega} y_1(\cdot, 0) v(\cdot, 0), \quad \forall v \in W_2^{1,1}(Q), \end{aligned} \quad (14)$$

$$\begin{aligned} \iint_{Q_2} y_2 v_t - \iint_{Q_2} \nabla y_2 \cdot \nabla v - \frac{1}{\gamma} \iint_{Q_2} p_2 v &= \iint_{Q_2} f v + \int_{\Omega} y_2(\cdot, T) v(\cdot, T) \\ &\quad - \int_{\Omega} y_2(\cdot, T_1^r) v(\cdot, T_1^r), \quad \forall v \in W_2^{1,1}(Q). \end{aligned} \quad (15)$$

Considering $y = y_1$ in Q_1 , $y = y_2$ in Q_2 , $y_1(\cdot, T_1^r) = y_2(\cdot, T_1^r)$ and summing (14) and (15) together, we get a weak formulation for equations of y that is equivalent to (12). An analogous argument leads to, given $p = p_1$ in Q_1 and $p = p_2$ in Q_2 , the weak formulation for equations of p in the coupled system is equivalent to (13).

(II) We then consider the overlapping decomposition in time domain, i.e., $T_1^r > T_2^l$. Denote by $\tilde{y} = y_1 - y_2$ and $\tilde{p} = p_1 - p_2$ on $Q^* = Q_1 \cap Q_2$, then

$$\begin{cases} -\partial_t \tilde{y} + \Delta \tilde{y} - \tilde{p}/\gamma = 0 & \text{in } Q^*, \\ \tilde{y} = 0 & \text{on } \Sigma, \quad \tilde{y}(\cdot, T_2^l) = 0 & \text{in } \Omega, \\ \partial_t \tilde{p} + \Delta \tilde{p} + \tilde{y} = 0 & \text{in } Q^*, \\ \tilde{p} = 0 & \text{on } \Sigma, \quad \tilde{p}(\cdot, T_1^r) = 0 & \text{in } \Omega. \end{cases} \quad (16)$$

We first show $\tilde{y}, \tilde{p} \in W(T_2^l, T_1^r; Q^*)$ is identically zero. In the equation of \tilde{y} , taking \tilde{p} as the test function, we have

$$-\iint_{Q^*} \partial_t \tilde{y} \tilde{p} + \iint_{Q^*} \Delta \tilde{y} \tilde{p} - \frac{1}{\gamma} \iint_{Q^*} \tilde{p}^2 = 0. \quad (17)$$

After integration by parts, it can be written as

$$\iint_{Q^*} \tilde{y} \partial_t \tilde{p} - \int_{\Omega} [\tilde{y}(T_1^r) \tilde{p}(T_1^r) - \tilde{y}(T_2^l) \tilde{p}(T_2^l)] - \iint_{Q^*} \nabla \tilde{y} \nabla \tilde{p} + \int_{\Sigma} \frac{\partial \tilde{y}}{\partial n} \tilde{p} - \frac{1}{\gamma} \iint_{Q^*} \tilde{p}^2 = 0. \quad (18)$$

Analogously, we take \tilde{y} as the test function in the equation for \tilde{p} and get

$$\iint_{Q^*} \partial_t \tilde{p} \tilde{y} + \iint_{Q^*} \Delta \tilde{p} \tilde{y} + \iint_{Q^*} \tilde{y}^2 = 0. \quad (19)$$

Integration by parts on the second term leads to

$$\iint_{Q^*} \partial_t \tilde{p} \tilde{y} - \iint_{Q^*} \nabla \tilde{p} \nabla \tilde{y} + \int_{\Sigma} \frac{\partial \tilde{p}}{\partial n} \tilde{y} + \iint_{Q^*} \tilde{y}^2 = 0. \quad (20)$$

Subtracting (18) from (20) and considering the boundary and initial conditions, we have

$$\iint_{Q^*} \tilde{y}^2 + \frac{1}{\gamma} \iint_{Q^*} \tilde{p}^2 = 0. \quad (21)$$

It indicates that \tilde{y} and \tilde{p} are identically zero on Q^* , therefore, $y_1 = y_2$ and $p_1 = p_2$ on Q^* .

Therefore, the overlapping case can be regarded as a decomposition of two non-overlapping subdomains $\Omega \times (0, T_1^r)$ and $\Omega \times (T_1^r, T)$. Similar to the case (I), it is easy to show that the hybrid formulation (9) and (10) is equivalent to the optimality system (3). This completes the proof. \square

The above hybrid formulation (9)-(10) can be easily extended to the multi-domain case, which motivates our following introduced iterative domain decomposition algorithms in the next section.

4. One-level Time Domain Decomposition Algorithms with Convergence Analysis

To take advantage of massively parallel computers, we need to consider many subdomains so that the computational task for solving the original optimality system can be allocated to multiple processors and be solved in a parallel manner. In this paper, we consider a strip decomposition in the time domain. In nonoverlapping algorithms, we partition the time domain $[0, T]$ into K non-overlapping subdomains of

equal length. For overlapping algorithms, we extend each nonoverlapping subdomain to its neighboring subdomains with an overlap length $\delta > 0$. Denote by $Q_i = \Omega \times (T_i^l, T_i^r)$ the i -th subdomain. For the first and last subdomain, we will enforce the maximum length of interval by setting $T_1^l = 0$ and $T_K^r = T$. A typical scenario of 3 overlapping subdomains is illustrated in Fig. 1.

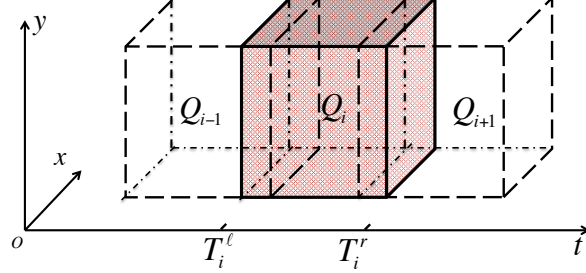


Figure 1: A schematic diagram of strip domain decomposition in time.

For simplicity of exposition, we will present our algorithms and convergence analysis for the case with only two subdomains (i.e., $K = 2$), but the generalization of our results to many subdomains are straightforward and the convergence will also be demonstrated by our numerical results. Suppose the state y_i^{k-1} and adjoint state p_i^{k-1} on Q_i ($i = 1, 2$) are given at the $(k-1)$ -th iteration, then y_i^k and p_i^k satisfy:

$$\begin{cases} -\partial_t y_1^k + \Delta y_1^k - p_1^k / \gamma = f & \text{in } Q_1, & y_1^k(\cdot, T_1^l) = y_0 & \text{in } \Omega, \\ \partial_t p_1^k + \Delta p_1^k + y_1^k = g & \text{in } Q_1, & p_1^k(\cdot, T_1^r) = p_2^{k-1}(T_1^r) & \text{in } \Omega, \end{cases} \quad (22)$$

$$\begin{cases} -\partial_t y_2^k + \Delta y_2^k - p_2^k / \gamma = f & \text{in } Q_2, & y_2^k(\cdot, T_2^l) = y_1^k(\cdot, T_2^l) & \text{in } \Omega, \\ \partial_t p_2^k + \Delta p_2^k + y_2^k = g & \text{in } Q_2, & p_2^k(\cdot, T_2^r) = 0 & \text{in } \Omega, \end{cases}$$

where the original homogeneous boundary conditions $y_1 = 0, p_1 = 0, y_2 = 0$, and $p_2 = 0$ hold for all k .

When $\star = (k-1)$ in (22), we obtain an *additive* Schwarz algorithm. It solves the optimality system in a highly parallel manner. Since interface conditions come from the previous iteration, the new approximations y_i^k and p_i^k on each subdomain Q_i can be computed concurrently.

When $\star = k$ in (22), we get a *multiplicative* Schwarz algorithm. It solves the optimality system by sequentially updating the approximations on subdomains in a prescribed order. In fact, the latest interface value $y_{i-1}^k(\cdot, T_{i-1}^r)$ from the preceding subdomain Q_{i-1} is immediately used in updating y_i^k and p_i^k on the subdomain Q_i . Therefore, the multiplicative algorithm runs sequentially in nature. As a standard strategy, its parallelizability can be greatly improved by grouping the subdomains into different colors. The subdomains with the same color do not intersect with each other and, thus, can be solved simultaneously. Indeed, since we consider the decomposition in the 1D time direction, two colors are sufficient to group all the strips.

We will study four different domain decomposition algorithms: (1) additive Schwarz with no overlap (ASN); (2) multiplicative Schwarz iterations with no overlap (MSN); (3) additive Schwarz iterations with overlap (ASO); (4) multiplicative Schwarz iterations with overlap (MSO). With numerical tests, we will compare all of them (1-level and 2-level) as both stand-alone iterative solvers and preconditioners of GMRES.

Next, we analyze the convergence of the non-overlapping algorithms, ASN and MSN, respectively. In the following, we assume the case with two subdomains. Denote $T_1 = T_2^l = T_1^r \in (0, T)$ and the global time interval $[0, T]$ is then divided into two nonoverlapping sub-intervals $[0, T_1]$ and $[T_1, T]$. Upon a semi-discretization of the Laplacian operator with a second-order central difference scheme in space, the ASN

algorithm based on (22) iterates concurrently according to the following two coupled ODEs

$$\begin{cases} -\partial_t y_1^k + \Delta_h y_1^k - p_1^k/\gamma = f_1 & \text{in } Q_1, & y_1^k(0) = y_0 & \text{in } \Omega, \\ \partial_t p_1^k + \Delta_h p_1^k + y_1^k = g_1 & \text{in } Q_1, & p_1^k(T_1) = p_2^{k-1}(T_1) & \text{in } \Omega, \end{cases} \quad (23)$$

$$\begin{cases} -\partial_t y_2^k + \Delta_h y_2^k - p_2^k/\gamma = f_2 & \text{in } Q_2, & y_2^k(T_1) = y_1^{k-1}(T_1) & \text{in } \Omega, \\ \partial_t p_2^k + \Delta_h p_2^k + y_2^k = g_2 & \text{in } Q_2, & p_2^k(T) = 0 & \text{in } \Omega, \end{cases} \quad (24)$$

where $y_1^k(t), y_2^k(t), p_1^k(t), p_2^k(t), f_1(t), f_2(t), g_1(t), g_2(t)$ are the corresponding spatially discretized vector functions and Δ_h is a discrete Laplacian operator associated with the Dirichlet boundary conditions. It is worthwhile to point out that, although we use the discrete Laplacian with a 5-point stencil central difference in the full discrete scheme (4-7), our following analysis also applies to other different discretization of the Laplacian operator. The initial and ending vectors $y_1^{k-1}(T_1), p_2^{k-1}(T_1)$ of $(k-1)$ -th iteration are assumed to be given or already obtained. In the following theorem we proved the convergence of the above semi-discretized ASN algorithm, by treating time in a continuous manner.

Theorem 4.1. *The ASN algorithm given by the iterations (23-24) is convergent.*

Proof. According to Theorem 3.1, subtracting (23-24) from the corresponding semi-discretized version of (9-10) leads to the following coupled system of the approximation error functions

$$\begin{cases} -\partial_t e_1^k + \Delta_h e_1^k - w_1^k/\gamma = 0 & \text{in } Q_1, & e_1^k(0) = 0 & \text{in } \Omega, \\ \partial_t w_1^k + \Delta_h w_1^k + e_1^k = 0 & \text{in } Q_1, & w_1^k(T_1) = w_2^{k-1}(T_1) & \text{in } \Omega, \end{cases} \quad (25)$$

$$\begin{cases} -\partial_t e_2^k + \Delta_h e_2^k - w_2^k/\gamma = 0 & \text{in } Q_2, & e_2^k(T_1) = e_1^{k-1}(T_1) & \text{in } \Omega, \\ \partial_t w_2^k + \Delta_h w_2^k + e_2^k = 0 & \text{in } Q_2, & w_2^k(T) = 0 & \text{in } \Omega, \end{cases} \quad (26)$$

where $e_1^k(t) = y_1^k(t) - y_1(t)$, $e_2^k(t) = y_2^k(t) - y_2(t)$, $w_1^k(t) = p_1^k(t) - p_1(t)$, $w_2^k(t) = p_2^k(t) - p_2(t)$ are the error vector functions of the state y and adjoint state p on each sub-interval. If we can show that both $e_1^k(T_1)$ and $w_2^k(T_1)$ converge to zero as k goes to infinity, then the uniqueness of the solution to both (25) and (26) obviously implies all error vector functions over each sub-interval also converge to zero as k goes to infinity.

In (25), it follows from multiplying from the left side of the first equation with $(e_1^k)^\top$, the transpose of (e_1^k) , and the second one with $(w_1^k)^\top$, respectively,

$$\begin{cases} -\frac{1}{2}\partial_t[(e_1^k)^\top e_1^k] + (e_1^k)^\top \Delta_h e_1^k - (e_1^k)^\top w_1^k/\gamma = 0 & \text{in } Q_1, & e_1^k(0) = 0 & \text{in } \Omega, \\ \frac{1}{2}\partial_t[(w_1^k)^\top w_1^k] + (w_1^k)^\top \Delta_h w_1^k + (w_1^k)^\top e_1^k = 0 & \text{in } Q_1, & w_1^k(T_1) = w_2^{k-1}(T_1) & \text{in } \Omega. \end{cases} \quad (27)$$

Since $(e_1^k)^\top w_1^k = (w_1^k)^\top e_1^k$, the addition of the first equation multiplied with γ and the second one leads to

$$\begin{cases} -\frac{1}{2}\partial_t \gamma[(e_1^k)^\top e_1^k] + \frac{1}{2}\partial_t[(w_1^k)^\top w_1^k] + \gamma(e_1^k)^\top \Delta_h e_1^k + (w_1^k)^\top \Delta_h w_1^k = 0 & \text{in } Q_1, \\ e_1^k(0) = 0, & w_1^k(T_1) = w_2^{k-1}(T_1) & \text{in } \Omega. \end{cases} \quad (28)$$

Applying the Fundamental Theorems of Calculus, a straightforward integration of the above equation from

0 to T_1 gives

$$\left\{ \begin{array}{l} \frac{1}{2}\gamma[(e_1^k(0))^\top e_1^k(0) - (e_1^k(T_1))^\top e_1^k(T_1)] + \frac{1}{2}[(w_1^k(T_1))^\top w_1^k(T_1) - (w_1^k(0))^\top w_1^k(0)] + \\ \int_0^{T_1} [\gamma(e_1^k)^\top \Delta_h e_1^k] dt + \int_0^{T_1} [(w_1^k)^\top \Delta_h w_1^k] dt = 0 \text{ in } Q_1, \\ e_1^k(0) = 0, \quad w_1^k(T_1) = w_2^{k-1}(T_1) \text{ in } \Omega. \end{array} \right. \quad (29)$$

It then follows by enforcing the given end-point conditions and rearranging the terms (using 2-norm $\|\cdot\|$)

$$\gamma\|e_1^k(T_1)\|^2 = \|w_2^{k-1}(T_1)\|^2 - \|w_1^k(0)\|^2 - \int_0^{T_1} 2\gamma\|\nabla_h e_1^k\|^2 dt - \int_0^{T_1} 2\|\nabla_h w_1^k\|^2 dt, \quad (30)$$

where we have used the fact (based on the discrete version of integration by parts [36])

$$z^\top (-\Delta_h) z = \langle z, (-\Delta_h) z \rangle = \langle \nabla_h z, \nabla_h z \rangle = \|\nabla_h z\|^2.$$

Similarly in (26), with an integration from T_1 to T , we will get

$$\left\{ \begin{array}{l} \frac{1}{2}\gamma[(e_2^k(T_1))^\top e_2^k(T_1) - (e_2^k(T))^\top e_2^k(T)] + \frac{1}{2}[(w_2^k(T))^\top w_2^k(T) - (w_2^k(T_1))^\top w_2^k(T_1)] + \\ \int_{T_1}^T [\gamma(e_2^k)^\top \Delta_h e_2^k] dt + \int_{T_1}^T [(w_2^k)^\top \Delta_h w_2^k] dt = 0 \text{ in } Q_2, \\ e_2^k(T_1) = e_1^{k-1}(T_1), \quad w_2^k(T) = 0 \text{ in } \Omega, \end{array} \right. \quad (31)$$

from which we obtain

$$\|w_2^k(T_1)\|^2 = \gamma\|e_1^{k-1}(T_1)\|^2 - \gamma\|e_2^k(T)\|^2 - \int_{T_1}^T 2\gamma\|\nabla_h e_2^k\|^2 dt - \int_{T_1}^T 2\|\nabla_h w_2^k\|^2 dt. \quad (32)$$

Add (30) and (32) together to get

$$\begin{aligned} \gamma\|e_1^k(T_1)\|^2 + \|w_2^k(T_1)\|^2 &= \gamma\|e_1^{k-1}(T_1)\|^2 + \|w_2^{k-1}(T_1)\|^2 - \|w_1^k(0)\|^2 - \gamma\|e_2^k(T)\|^2 \\ &\quad - \int_0^{T_1} 2\gamma\|\nabla_h e_1^k\|^2 dt - \int_0^{T_1} 2\|\nabla_h w_1^k\|^2 dt - \int_{T_1}^T 2\gamma\|\nabla_h e_2^k\|^2 dt - \int_{T_1}^T 2\|\nabla_h w_2^k\|^2 dt \\ &\leq \gamma\|e_1^{k-1}(T_1)\|^2 + \|w_2^{k-1}(T_1)\|^2 - \|w_1^k(0)\|^2 - \gamma\|e_2^k(T)\|^2 \\ &\quad - \int_0^{T_1} 2\gamma c\|e_1^k\|^2 dt - \int_0^{T_1} 2c\|w_1^k\|^2 dt - \int_{T_1}^T 2\gamma c\|e_2^k\|^2 dt - \int_{T_1}^T 2c\|w_2^k\|^2 dt \\ &< \gamma\|e_1^{k-1}(T_1)\|^2 + \|w_2^{k-1}(T_1)\|^2, \end{aligned} \quad (33)$$

where in the second last step we used the following inequalities (based on the discrete version of Poincaré inequality)

$$\|e_1^k\|^2 \leq (1/c)\|\nabla_h e_1^k\|^2, \|e_2^k\|^2 \leq (1/c)\|\nabla_h e_2^k\|^2, \|w_1^k\|^2 \leq (1/c)\|\nabla_h w_1^k\|^2, \|w_2^k\|^2 \leq (1/c)\|\nabla_h w_2^k\|^2,$$

with a positive constant c independent of h , T and T_1 .

From (33), we can easily see that the sequence $\{\gamma\|e_1^k(T_1)\|^2 + \|w_2^k(T_1)\|^2\}_{k=0}^\infty$ is strictly decreasing unless $e_1^k \equiv 0$, $e_2^k \equiv 0$, $w_1^k \equiv 0$, and $w_2^k \equiv 0$ on the corresponding sub-intervals, which hence implies (note $\gamma > 0$)

$$\lim_{k \rightarrow \infty} \gamma\|e_1^k(T_1)\|^2 + \|w_2^k(T_1)\|^2 = 0.$$

Therefore, for any initial guess, both $e_1^k(T_1)$ and $w_2^k(T_1)$ converge to zero as $k \rightarrow \infty$. This completes the proof of the convergence of our ASN algorithm. \square

Repeating the above arguments, we can obtain the convergence results for the following semi-discretized

MSN algorithm:

$$\begin{cases} -\partial_t y_1^k + \Delta_h y_1^k - p_1^k/\gamma = f_1 & \text{in } Q_1, & y_1^k(0) = y_0 & \text{in } \Omega, \\ \partial_t p_1^k + \Delta_h p_1^k + y_1^k = g_1 & \text{in } Q_1, & p_1^k(T_1) = p_2^{k-1}(T_1) & \text{in } \Omega, \end{cases} \quad (34)$$

$$\begin{cases} -\partial_t y_2^k + \Delta_h y_2^k - p_2^k/\gamma = f_2 & \text{in } Q_2, & y_2^k(T_1) = y_1^k(T_1) & \text{in } \Omega, \\ \partial_t p_2^k + \Delta_h p_2^k + y_2^k = g_2 & \text{in } Q_2, & p_2^k(T) = 0 & \text{in } \Omega. \end{cases} \quad (35)$$

To illustrate the convergence difference between ASN and MSN algorithm, we briefly sketch the proof below.

Theorem 4.2. *The MSN algorithm given by the iterations (34-35) is convergent.*

Proof. According to Theorem 3.1, subtracting (34-35) from the corresponding semi-discretized version of (9-10) leads to the following coupled system of the approximation error functions

$$\begin{cases} -\partial_t e_1^k + \Delta_h e_1^k - w_1^k/\gamma = 0 & \text{in } Q_1, & e_1^k(0) = 0 & \text{in } \Omega, \\ \partial_t w_1^k + \Delta_h w_1^k + e_1^k = 0 & \text{in } Q_1, & w_1^k(T_1) = w_2^{k-1}(T_1) & \text{in } \Omega, \end{cases} \quad (36)$$

$$\begin{cases} -\partial_t e_2^k + \Delta_h e_2^k - w_2^k/\gamma = 0 & \text{in } Q_2, & e_2^k(T_1) = e_1^k(T_1) & \text{in } \Omega, \\ \partial_t w_2^k + \Delta_h w_2^k + e_2^k = 0 & \text{in } Q_2, & w_2^k(T) = 0 & \text{in } \Omega. \end{cases} \quad (37)$$

In (36), it follows from the same procedure that

$$\gamma \|e_1^k(T_1)\|^2 = \|w_2^{k-1}(T_1)\|^2 - \|w_1^k(0)\|^2 - \int_0^{T_1} 2\gamma \|\nabla_h e_1^k\|^2 dt - \int_0^{T_1} 2\|\nabla_h w_1^k\|^2 dt. \quad (38)$$

Similarly, in (37), we can obtain the slightly different

$$\|w_2^k(T_1)\|^2 = \gamma \|e_1^k(T_1)\|^2 - \gamma \|e_2^k(T)\|^2 - \int_{T_1}^T 2\gamma \|\nabla_h e_2^k\|^2 dt - \int_{T_1}^T 2\|\nabla_h w_2^k\|^2 dt. \quad (39)$$

Adding (38) and (39) together, we have

$$\begin{aligned} \|w_2^k(T_1)\|^2 &= \|w_2^{k-1}(T_1)\|^2 - \|w_1^k(0)\|^2 - \gamma \|e_2^k(T)\|^2 \\ &\quad - \int_0^{T_1} 2\gamma \|\nabla_h e_1^k\|^2 dt - \int_0^{T_1} 2\|\nabla_h w_1^k\|^2 dt - \int_{T_1}^T 2\gamma \|\nabla_h e_2^k\|^2 dt - \int_{T_1}^T 2\|\nabla_h w_2^k\|^2 dt \\ &\leq \|w_2^{k-1}(T_1)\|^2 - \|w_1^k(0)\|^2 - \gamma \|e_2^k(T)\|^2 \\ &\quad - \int_0^{T_1} 2\gamma c \|e_1^k\|^2 dt - \int_0^{T_1} 2c \|w_1^k\|^2 dt - \int_{T_1}^T 2\gamma c \|e_2^k\|^2 dt - \int_{T_1}^T 2c \|w_2^k\|^2 dt \\ &< \|w_2^{k-1}(T_1)\|^2, \end{aligned} \quad (40)$$

from which we can easily see that the sequence $\{\|w_2^k(T_1)\|^2\}_{k=0}^\infty$ is strictly decreasing and hence there holds

$$\lim_{k \rightarrow \infty} \|w_2^k(T_1)\|^2 = 0.$$

Then it follows clearly from (38) that

$$\lim_{k \rightarrow \infty} \|e_1^k(T_1)\|^2 = 0.$$

Therefore, for any initial guess, both $e_1^k(T_1)$ and $w_2^k(T_1)$ converge to zero as $k \rightarrow \infty$. This completes the proof of the convergence of our MSN algorithm. \square

The above conclusions are counter-intuitively surprising in the view of the well-recognized fact that a standard ASN algorithm without overlap for elliptic boundary value problems is not convergent as a stand-

alone iterative solver [37, 38]. Nonetheless, the above convergence result does not provide any explicit estimate of the convergence rate of the ASN algorithm, which will be carried out further in the following by making use of more detailed information contained in (33). Let $y^k(t)$ be the approximation of $y(t)$ at k -th iteration by gluing y_1^k and y_2^k together (taking the value of y_1^k at T_1). Similarly, let $p^k(t)$ be the approximation of $p(t)$ at k -th iteration by gluing p_1^k and p_2^k together (taking the value of p_2^k at T_1). Define the global error vector functions $e^k(t) = y^k(t) - y(t)$ and $w^k(t) = p^k(t) - p(t)$. Then there holds (in the sense of Lebesgue integral)

$$\int_0^{T_1} 2\gamma c \|e_1^k\|^2 dt + \int_{T_1}^T 2\gamma c \|e_2^k\|^2 dt = 2c \int_0^T \gamma \|e^k\|^2 dt$$

and

$$\int_0^{T_1} 2c \|w_1^k\|^2 dt + \int_{T_1}^T 2c \|w_2^k\|^2 dt = 2c \int_0^T \|w^k\|^2 dt.$$

Let $z^k(t) := \gamma \|e^k(t)\|^2 + \|w^k(t)\|^2$. Then by definition we have $z^k(T_1) = \gamma \|e_1^k(T_1)\|^2 + \|w_2^k(T_1)\|^2$. By using this new vector function $z^k(t)$ and dropping the non-positive terms $(-\|w_1^k(0)\|^2 - \gamma \|e_2^k(T)\|^2)$, it follows from (33) that

$$z^k(T_1) \leq z^{k-1}(T_1) - 2c \int_0^T z^k(t) dt \quad (41)$$

holds for any fixed $T_1 \in (0, T)$. Considering that $z^k(0) = \|w_1^k(0)\|^2$ and $z^k(T) = \gamma \|e_2^k(T)\|^2$ are the approximation errors at both end points, we assume there holds $z^k(t) \geq \beta_k z^k(T_1)$ for some $\beta_k > 0$, i.e., the global minimal approximation error is roughly bounded below by the local error at $t = T_1$. With this assumption and in view of (41), we can arrive

$$z^k(T_1) \leq z^{k-1}(T_1) - 2c\beta_k T z^k(T_1), \quad (42)$$

which clearly implies

$$z^k(T_1) \leq \frac{1}{1+2c\beta_k T} z^{k-1}(T_1), \quad (43)$$

where β_k is expected to be dependent on k , T , and the problem. In view of (43), the convergence rate of the ASN algorithm is mainly determined by the value of β_k . In particular, numerical simulations show that the convergence rate of 1-level ASN algorithm gets worse as the approximation errors become smaller. This can be explained by the easy-to-check fact that β_k indeed decreases to zero as k goes to infinity, which can also be easily seen from (41) by observing $\int_0^T z^k(t) dt \rightarrow 0$ as k goes to infinity. Hence, from theoretical point of view, the 1-level ASN algorithm may converge very slow, especially when the spatial operator has a zero eigenvalue, which was also pointed out in [16]. For our considered spatially uniform elliptic problems, the obtained convergence rates are, however, very satisfactory in numerical.

The convergence analysis of the ASO and MSO algorithms will require some different proving techniques, which are not carried out in current paper. However, we numerically demonstrate the convergence of both one-level and two-level ASO and MSO algorithms as stand-alone solvers and preconditioners, respectively. According to our simulations, the ASO and MSO algorithms converge slightly faster than the ASN and MSN algorithms, respectively. It is well-known that adding a suitable overlap (as in ASO and MSO) in domain decomposition algorithms often leads to better convergence rate than nonoverlapping ones. Therefore, it is reasonable for us to believe that the proved convergence of ASN and MSN algorithms *nominally* implies (but not rigorously prove) the convergence of both ASO and MSO algorithms. Further convergence analysis of the ASO and MSO algorithms (including the effects of the size of overlap) will be investigated in future.

5. Two-level Time Domain Decomposition Algorithms with Many Subdomains

The above introduced one-level domain decomposition algorithms works reasonably well when the number of subdomains K is not large. However, existing theoretical and numerical results (see also Section 6) indicate that one-level domain decomposition methods based only on local subdomain solves are not scalable with respect to the number of subdomains K , which can be resolved by using a two-level algorithm in which the current one-level parallel subdomain iterations are corrected globally by solving an appropriately selected coarse grid problem with the size of order $O(K)$ [23]. Such two-level domain decomposition algorithms based on coarse grid corrections can be viewed as a variant of the standard two-level multigrid method, where the one-level parallel subdomain iterations function as an advanced smoother [39, 40]. Nevertheless, the coarse grid size in the context of two-level domain decomposition algorithms is usually significantly smaller than that of a two-level multigrid method based on halving the mesh sizes. In general, the successful construction of an effective and efficient coarse grid space is a very dedicated task that often requires many heuristic numerical investigations by trial and error [41]. In this section, we briefly describe our coarse grid spaces based on the ones discussed in [26, 27], which seem to be quite effective according to our following numerical experiments in Section 6.

We first consider the nonoverlapping cases in MSN and ASN algorithms. Although it is more convenient to conduct theoretical analysis in the continuous setting, here we choose to illustrate our algorithms in its discrete formulation as this is most faithful replication of the actual implementation in our codes. Assume $(N - 1)$ is an integer multiple of K . Let the global set of time grid points $\{0 = t_0 < t_1 < \dots < t_{N-1} < t_N = T\}$ be decomposed into K nonoverlapping subsets of equal size. The following Fig. 2 depicts a typical nonoverlapping decomposition scheme with $N = 17$ and $K = 4$, where the boundary nodes $t_0 = 0$ and $t_N = T$ will be absorbed into the right hand side during the computation. Notice that each subdomain will need to receive two boundary nodes approximations from its neighboring subdomains, e.g., subdomain 2 ($\{t_5, t_6, t_7, t_8\}$) treats the approximations at nodes t_4 and t_9 as virtual boundary nodes.

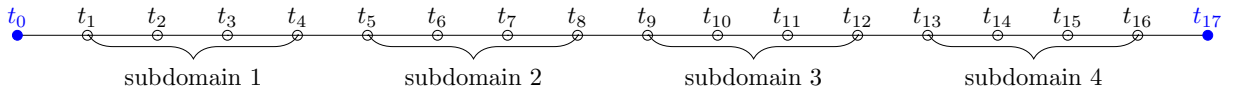


Figure 2: The nonoverlapping decomposition of all time grid points (empty circles) with $N = 17$ and $K = 4$.

Following the approach in [26, 27], we propose to choose the coarse grid nodes ($\{t_1, t_4, t_5, t_8, t_9, t_{12}, t_{13}, t_{16}\}$) as described in the following Fig. 3, where the adjacent grid nodes connecting two subdomains are selected as coarse grid nodes.

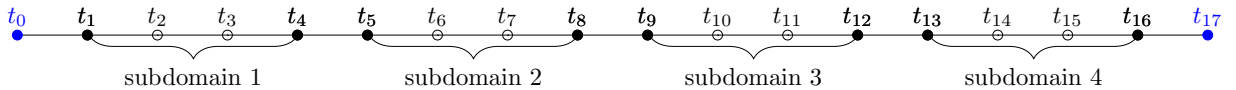


Figure 3: The choice of coarse grid nodes (filled circles) among fine grid points (all circles) with $N = 17$ and $K = 4$.

After the coarse grid nodes are determined, we follow the standard procedure to build the corresponding extension operator E based on linear interpolation and the restriction operator R as the transpose of E after row-normalization. Taking the coarse and fine grid nodes shown in Fig. 3 as an example, the corresponding extension operator $E \in \mathbb{R}^{16 \times 8}$ that interpolates a coarse grid approximation to a fine grid approximation

has the following matrix formulation (transposed for economical exposition)

$$E^\top = \begin{bmatrix} 1 & \frac{2}{3} & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{3} & \frac{2}{3} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \frac{2}{3} & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & \frac{2}{3} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \frac{2}{3} & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & \frac{2}{3} & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \frac{2}{3} & \frac{1}{3} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & \frac{2}{3} & 1 \end{bmatrix}_{8 \times 16}. \quad (44)$$

The corresponding restriction operator R is given by $R = \frac{1}{2}E^\top$, with $\frac{1}{2}$ being the normalization factor calculated from the row sum of E^\top . With both E and R in hands, one can algebraically setup the coarse grid coefficient matrix through the Galerkin projection

$$L_c = RL_hE, \quad (45)$$

where the coarse grid matrix L_c has a much smaller dimension of order $O(K)$. Similar as the geometric multi-grid method, it is also possible to geometrically construct the coarse grid coefficient matrix by re-discretizing the original continuous problem with a chosen coarse mesh (needs to be uniform in our current finite difference scheme). Our preliminary numerical tests show that such a geometrical approach of constructing the coarse grid matrix and the corresponding extension and restriction operators delivers less robust convergence rates than the above algebraic approach. A 2-level ASN or MSN algorithm is obtained by complementing the above 1-level ASN or MSN algorithms (either as solver or preconditioner) with one step of coarse grid correction based on the residual equation. Suppose we have obtained a 1-level global solution approximation $w_h^{(1)}$ from each iteration of 1-level ASN or MSN algorithms, the corresponding improved 2-level approximation $w_h^{(2)}$ can be calculated according to the correction

$$w_h^{(2)} = w_h^{(1)} + EL_c^{-1}R(b_h - w_h^{(1)}), \quad (46)$$

where L_c^{-1} should be understood as one efficient coarse grid system solving with certain level of approximation errors. Here we in fact applied the coarse grid correction in a multiplicative manner [18]. It is worthwhile to mention that the extra cost of adding one coarse grid correction as in (46) is usually negligible compared to the overall computational costs, if the number of subdomains is far less than the total number of time grid points, i.e., $K \ll N$.

Next, we consider the overlapping cases in MSO and ASO algorithms. The following Fig. 4 depicts a typical overlapping decomposition scheme with $N = 17$ and $K = 4$, where each subdomain extends one grid node into its neighboring subdomains in view of the nonoverlapping decompositions shown in Fig. 2. Slightly different from the nonoverlapping cases, we choose the coarse grid nodes as described in Fig. 5, where one grid node centered at the overlap region is selected as a coarse grid node. Numerical tests show that the current choice gives scalable convergence rates, although other better choices may not be within our numerical trials. For instance, we also tested the classical choice of placing one coarse grid point into the center of each subdomain, which however does not provide very scalable convergence rates. Further discussion on the construction of better coarse grid spaces is beyond the scope of the current paper and is

left as our future research.

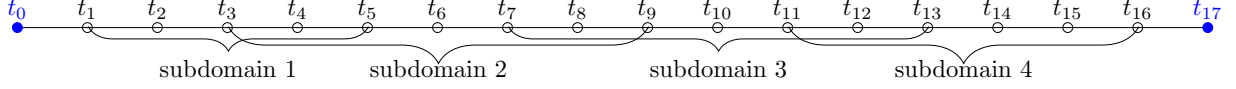


Figure 4: The overlapping decomposition of all time grid points (empty circles) with $N = 17$ and $K = 4$.

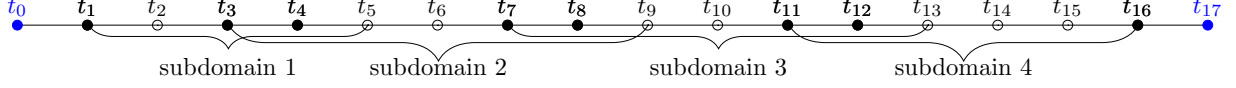


Figure 5: The choice of coarse grid nodes (filled circles) among fine grid points (all circles) with $N = 17$ and $K = 4$.

6. Numerical Examples

In this section, we provide numerical examples to demonstrate the effectiveness of our proposed methods. All simulations are implemented using MATLAB R2016a on a laptop PC with Intel(R) Core(TM) i3-3120M CPU@2.50GHz and 12GB RAM. We use the strip decomposition that divides the time interval $[0, T]$ into K uniform parts, thus the space-time domain is partitioned to K chunks. On each subdomain, we use the finite difference discretizations developed in [33], which provides a second-order accuracy in both space and time. The time step size τ is taken to be same as the spatial mesh size h . For 1D examples, the sub-domain systems are solved by sparse direct solvers, such as the backslash ‘mldivide’ solver of MATLAB. For 2D examples, the efficient semi-coarsening multigrid solver with one V-cycle [33] is utilized to approximately solve the sub-linear system on each subdomain. However, any other efficient iterative solvers, such as algebraic multigrid methods [42], are also applicable.

When overlapping methods are used, each subdomain is extended and overlaps its neighbors by a minimal overlap $\delta = \tau$ in the temporal direction. All iterative algorithms start with a random initial guess (use `rand` function in MATLAB). Notice that our domain decomposition algorithms can be used as stand-alone iterative solvers and preconditioners, respectively. We choose the standard stopping condition based on relative reduction of residual norms, i.e.,

$$\|r_k\|/\|r_0\| < 10^{-7},$$

where r_k is the global residual vector at k -th iteration and r_0 denotes the initial residual vector. We will use right-preconditioned GMRES and choose the same stopping criterion as the stand-alone solvers.

6.1. 1D Examples with Direct Subdomain Solvers

We first study one-dimensional examples to numerically verify our proposed methods. In 1D cases, all sub-domain systems are accurately solved up to a machine precision, with the approximation errors in solving sub-domain systems are far less than that caused by the domain decomposition iterations. This allows us to focus on evaluating the convergence performance of the domain decomposition algorithms themselves.

Example 1. Let $\Omega = (0, 1)$, $T = 4$, and $\gamma = 10^{-2}$. Choose f , g , and y_0 in (3) so that the exact solution is given by

$$y(x, t) = \cos(\pi t) \sin(\pi x_1) \text{ and } p(x, t) = \sin(\pi t) \sin(\pi x_1).$$

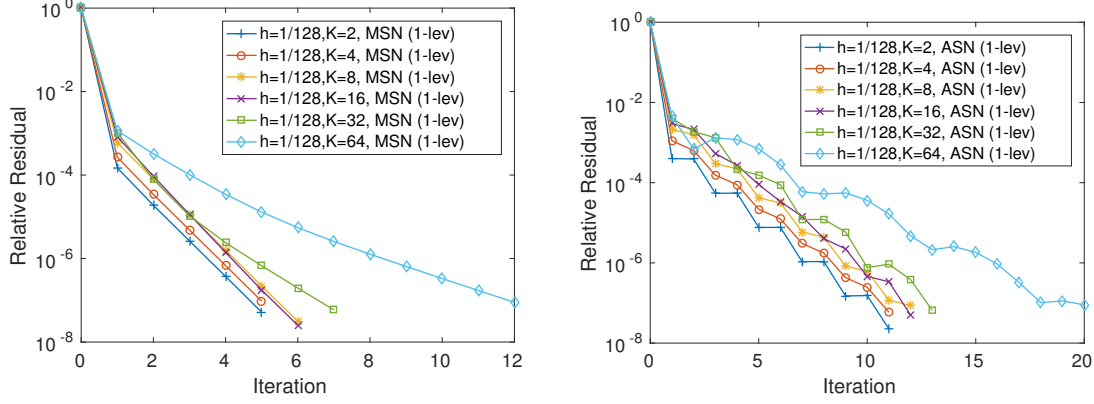


Figure 6: Convergence of 1-level MSN and ASN as stand-alone solvers.

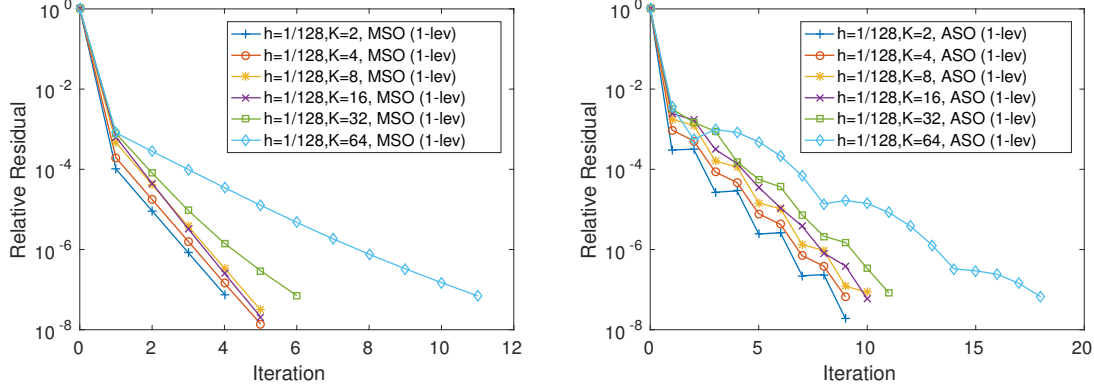


Figure 7: Convergence of 1-level MSO and ASO as stand-alone solvers.

In Fig. 6 and Fig. 7, we show the convergence of 1-level MSN, ASN, MSO, and ASO algorithms as stand-alone solvers, respectively. We observe that (i) the multiplicative solvers (MSN, MSO) require less iterations and therefore converge faster than the corresponding additive solvers (ASN, ASO); (ii) for both multiplicative and additive solvers, the corresponding overlapping solvers (MSO, ASO) have slightly faster convergence than the nonoverlapping ones (MSN, ASN) as the number of subdomains K is increased; Nevertheless, for all the solvers, the required iteration numbers show evident growth as the number of subdomains K increases up to 64, which is expected for such 1-level algorithms since more subdomains implies that more iterations are needed to exchange new approximations computed locally on one subdomain to all the other subdomains. In both figures, we observe that multiplicative algorithms converge monotonically, while additive algorithms show certain irregular non-monotonic decreasing in the computed residual norms. Such a difference can be briefly explained by a simple comparison of the proofs in both Theorem 4.1 and 4.2, where the MSN algorithm has strictly monotonic decrease in both error term $e_1^k(T_1)$ and $w_2^k(T_1)$ individually and the ASN algorithm only assures a strictly monotonic decrease in the weighted error term $(\gamma \|e_1^k(T_1)\|^2 + \|w_2^k(T_1)\|^2)$. Especially, when γ is small, its scaling effect on the global system with $1/\gamma$ may lead to our observed non-monotonic decreases of residual norms in the ASN algorithm as stand-alone solvers.

In order to obtain more robust convergence rate that is independent of the number of subdomains K , 2-level algorithms based on some appropriately chosen coarse space correction are often utilized to suppress

the increasing iteration numbers. In Fig. 8 and Fig. 9, we show the convergence of our implemented 2-level MSN, ASN, MSO, and ASO algorithms as stand-alone solvers, respectively. Here we used a standard coarse level space of a small size K based linear interpolation under the framework of Galerkin method, as discussed in [26, 27]. Compared to the above 1-level algorithms, we do observe that the required iteration numbers are almost independent of the increasing number of subdomains K , which is critical to the overall performance of the corresponding massively parallel implementations. Moreover, a coarse space correction step seems to be more effective in improving additive algorithms than the multiplicative ones.

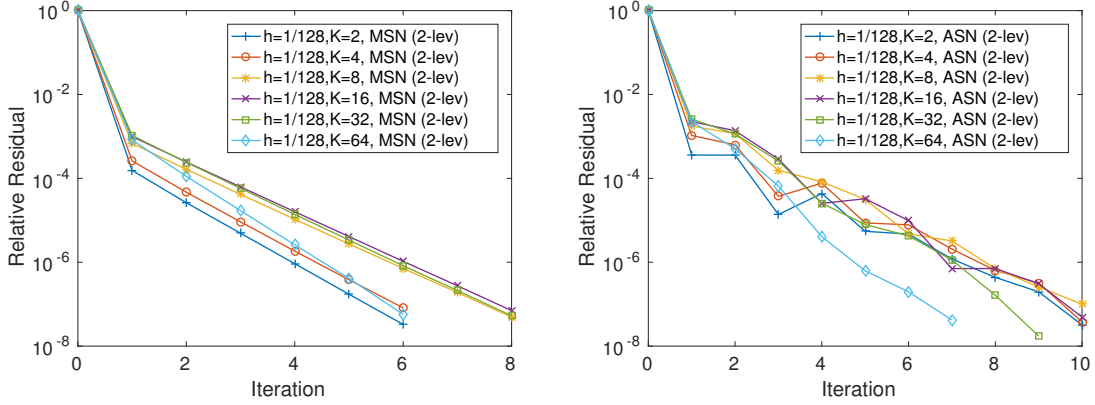


Figure 8: Convergence of 2-level MSN and ASN as stand-alone solvers.

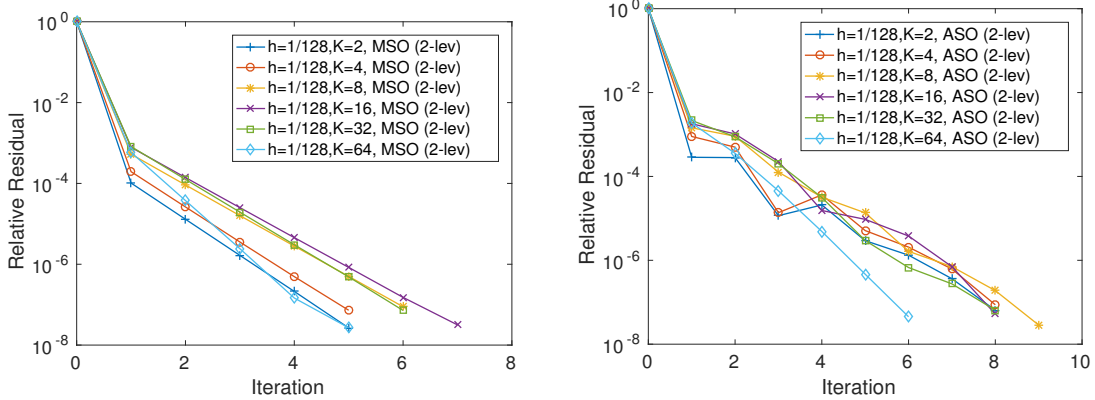


Figure 9: Convergence of 2-level MSO and ASO as stand-alone solvers.

Although theoretically sound, the above illustrated 1-level and 2-level algorithms as stand-alone solvers are seldom used in practice, since the overall convergence rate is very problem-dependent and sensitive to the choice of coarse level space. If the coarse space is not appropriately chosen, the stand-alone solver may become inefficient or even divergent. Therefore, these iterative algorithms are always used as effective preconditioners for some appropriate Krylov subspace methods, which often demonstrate more robust convergence rates than as stand-alone solvers. In Fig. 10, we plot the convergence of GMRES with no preconditioner (No Prec.) and with our proposed 1-level ASN preconditioner (with $K = 16$), respectively, for various refined mesh step sizes. Without any preconditioner, the GMRES demonstrates a clearly deteriorated convergence rate as the step size $h = \tau$ is halved, which in fact leads to a four times larger system of linear equations with

roughly four time larger condition number. However, the GMRES with 1-level ASN preconditioners shows an mesh-independent convergence, which attributes to the corresponding 1-level ASN stand-alone solver having a mesh-independent convergence rate. In contrast to the possible non-monotonic decay of residual norms in the ASN algorithm as a stand-alone solver, the preconditioned GMRES solver yields a robust monotonic decaying residual norms.

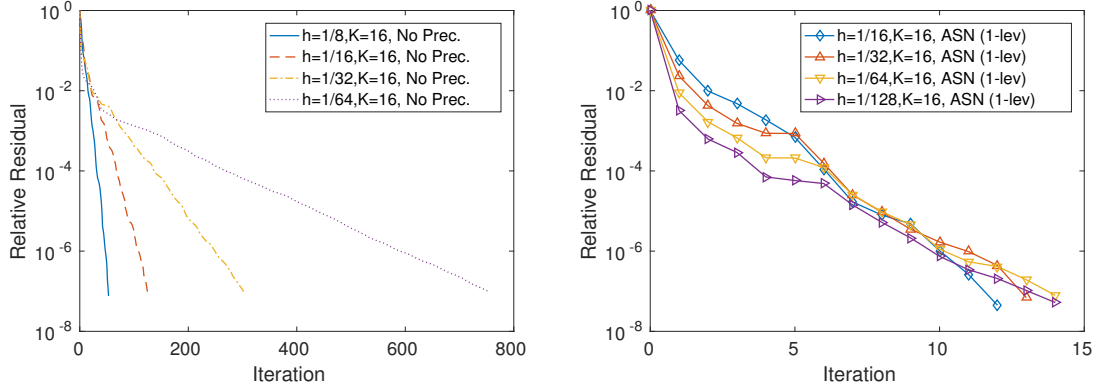


Figure 10: Convergence of GMRES with no preconditioner and 1-level ASN ($K = 16$) preconditioners, respectively.

We further examine the convergence performance of such 1-level domain decomposition preconditioners with respect to K . In Figures 11–12, we plot the convergence of right-preconditioned GMRES with 1-level MSN, ASN, MSO, and ASO algorithms as preconditioners for K different numbers of subdomains, respectively. Similar to what we have observed in 1-level stand-alone solvers, the convergence rates of the preconditioned GMRES with 1-level preconditioners are quite satisfactory for small K , but all of them become evidently slower as the number of subdomains K is increased.

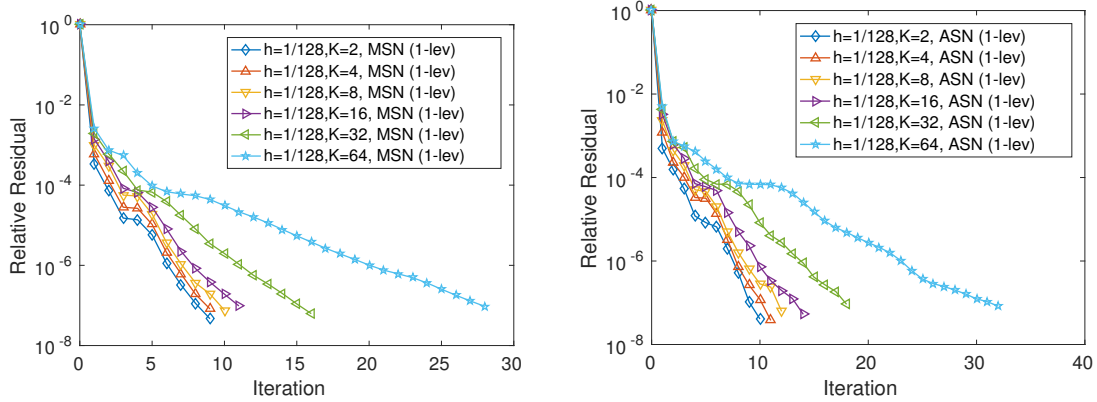


Figure 11: Convergence of GMRES with 1-level MSN and ASN preconditioners.

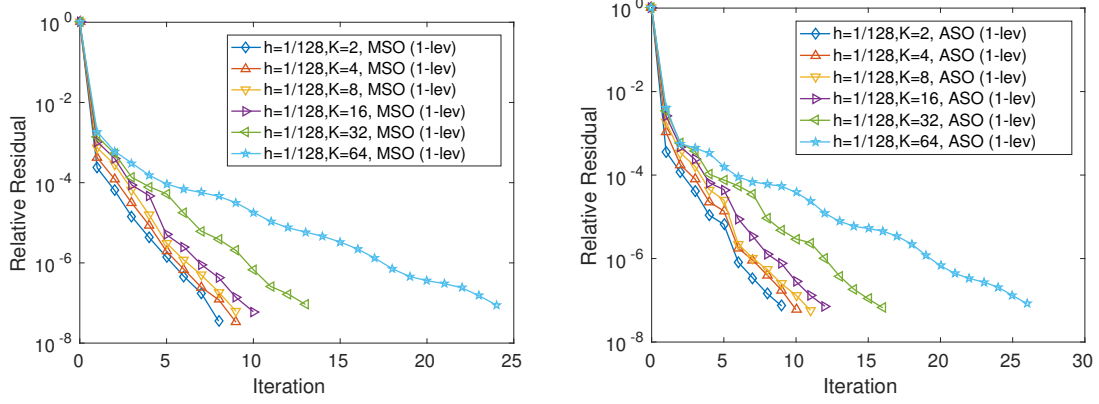


Figure 12: Convergence of GMRES with 1-level MSO and ASO preconditioners.

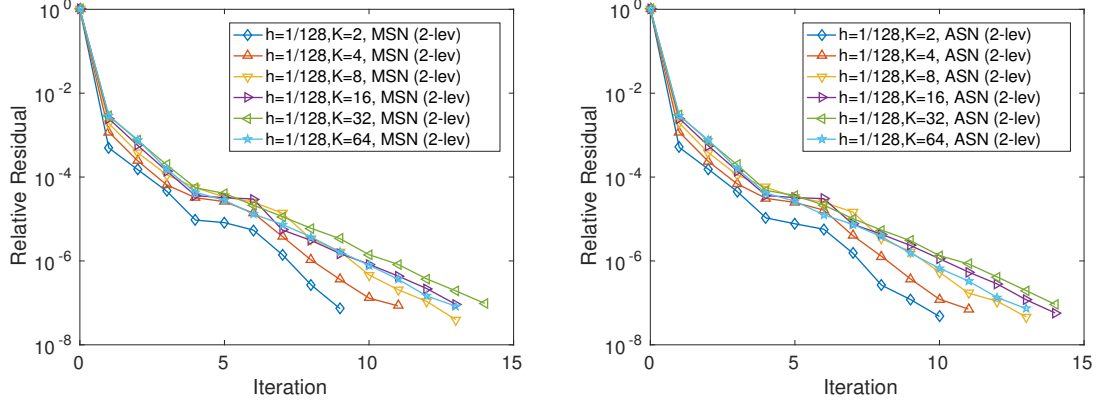


Figure 13: Convergence of GMRES with 2-level MSN and ASN preconditioners.

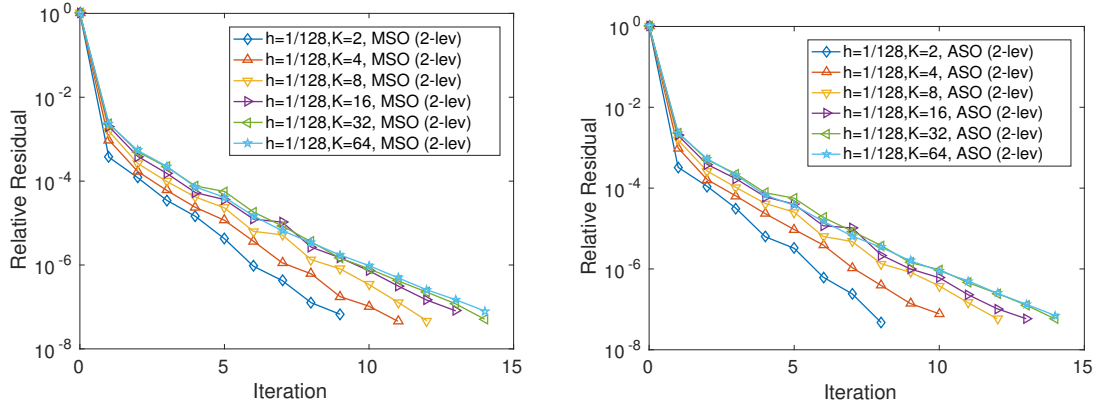


Figure 14: Convergence of GMRES with 2-level MSO and ASO preconditioners.

Based on the superior convergence performance of the above 2-level algorithms as stand-alone solvers, it is reasonable to expect such 2-level algorithms work very well as preconditioners for GMRES. As shown in Figures 13–14, the preconditioned GMRES with 2-level preconditioners indeed deliver more robust conver-

gence rate for large K . It is worthwhile to point out that 2-level preconditioners based on the coarse space correction may not necessarily take less iteration numbers than 1-level ones when K is relatively small. This can be explained by the possible large approximation errors introduced by a very coarse grid correction. However, 2-level preconditioners become significantly better than 1-level preconditioners for a large K .

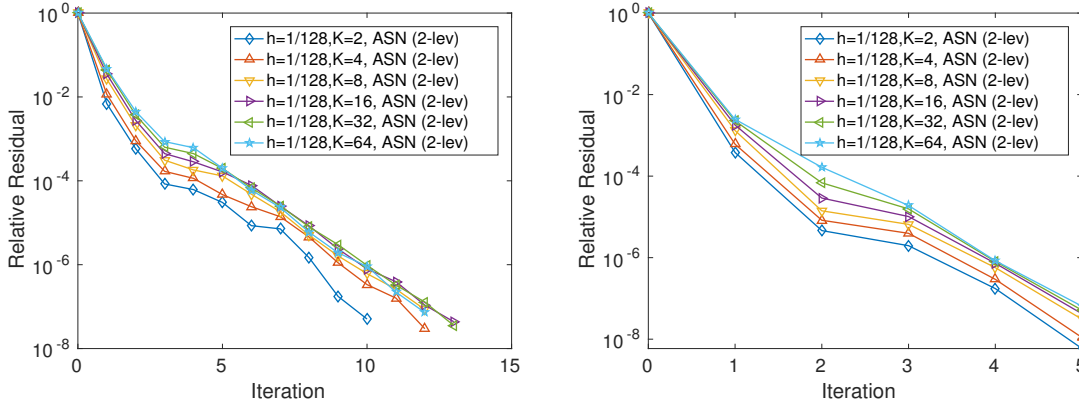


Figure 15: Convergence of GMRES with 2-level ASN preconditioners for $\gamma = 10^{-4}$ (left) and $\gamma = 10^{-6}$ (right).

The small parameter $\gamma > 0$ is usually chosen to be close to zero, which leads to the linear system (8) with a very large condition number. Consequently, it is often of great interests to obtain some efficient solvers with parameter-robust convergence rates. In Fig. 15, we show the convergence of GMRES with 2-level ASN preconditioners for a much smaller parameter $\gamma = 10^{-4}$ and $\gamma = 10^{-6}$, respectively. In view of Fig. 13 with $\gamma = 10^{-2}$, our 2-level ASN preconditioners seem to have very robust convergence rates with respect to varying γ .

6.2. 2D Examples with Multigrid Subdomain Solvers

Considering that time variable is only one dimensional, the application of our proposed algorithms to 2D or 3D problems is almost the same as the 1D problems. According to the convergence analysis in Section 4, our proposed algorithms are also convergent for such parabolic PDE control problems with two and three spatial dimensions. However, in 2D or 3D cases, subdomain systems become very expensive to be solved by sparse direct methods as we did in the above 1D cases. Hence, more efficient iterative solvers will be used for only approximately solving subdomain systems. More specifically, we will use the semi-coarsening multigrid solver with one V-cycle developed in [33] as subdomain system solver for our 2D tests. Notice that using more accurate subdomain system solvers will lead to improved convergence rate but may take higher computational costs. Based on our numerical simulations, one V-cycle multigrid iteration balances both accuracy and efficiency very well. In addition, the coarse space correction step in 2-level methods also requires fast iterative solvers, since it essentially solves a huge sparse linear system from an equivalent 3D discretization with only $O(K)$ grid points in the temporal direction. We will perform the coarse space correction by solving the coarse system approximately with the preconditioned bi-conjugate gradient stabilized method (BiCGStab) with an ILU(0) preconditioner (i.e., with 0 level of fill in), a tolerance of 10^{-4} and less than 200 iterations.

Example 2. Let $\Omega = (0, 1)^2$, $T = 4$, and $\gamma = 10^{-2}$. Choose f , g , and y_0 in (3) so that the exact solution

is given by

$$y(x, t) = \cos(\pi t) \sin(\pi x_1) \sin(\pi x_2) \text{ and } p(x, t) = \sin(\pi t) \sin(\pi x_1) \sin(\pi x_2).$$

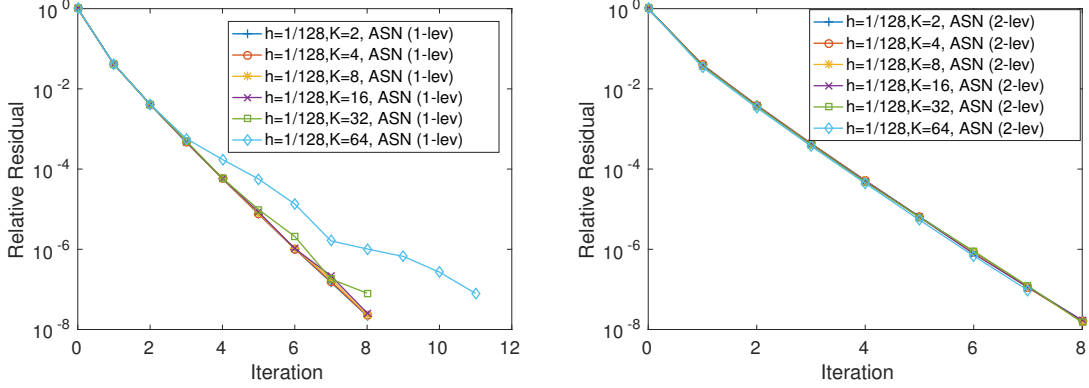


Figure 16: Convergence of 1-level and 2-level ASN as stand-alone solvers.

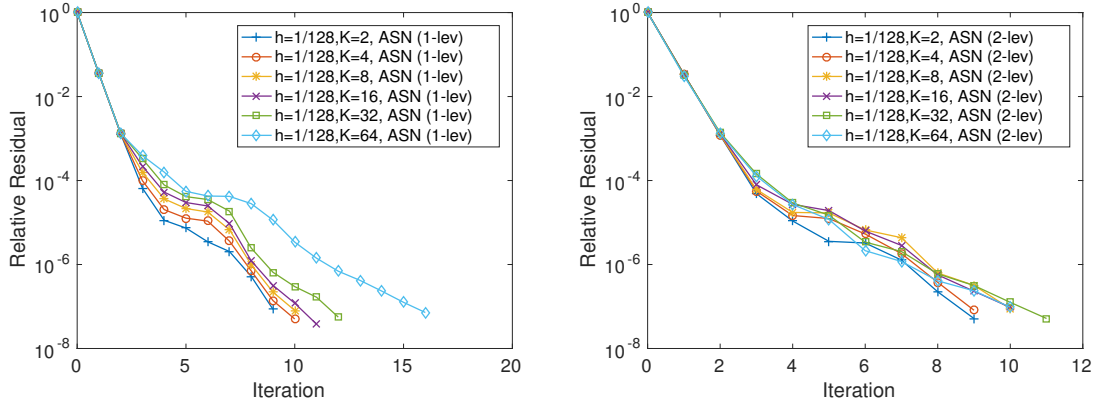


Figure 17: Convergence of GMRES with 1-level and 2-level ASN preconditioners.

In 2D cases, the convergence properties of all the proposed algorithms are essentially the same as the preceding 1D numerical results, hence we only report the simulation results of the ASN algorithm to keep our exposition concise. In Fig. 16, we show the convergence of 1-level and 2-level ASN algorithms as stand-alone solvers, respectively. In Fig. 17, we show the convergence of right-preconditioned GMRES with 1-level and 2-level ASN algorithm as preconditioners, respectively. Both figures are very similar to the 1D cases, where the 2-level ASN algorithm shows clearly improved convergence rate than the 1-level ASN algorithm with $K = 64$. Furthermore, the required number of iterations stay roughly the same as that in the 1D cases. In particular, the residual norms in 2D cases are monotonically decreasing. Notice that the 2D full discretization with $h = \tau = 1/128$ gives a large sparse linear system with about $128 \times 128 \times 128 \times 4 = 8,388,608$ unknowns.

7. Conclusions

We proposed, analyzed, and compared several different time domain decomposition algorithms for solving linear parabolic PDE control problems. These algorithms aim at solving the forward-and-backward optimality PDE system with parallel computers so that the one-shot method becomes more affordable in dealing

with 3D large-scale time-dependent PDE optimization problems. The proposed algorithms are shown to be convergent as stand-alone solvers and very effective as preconditioners of Krylov subspace methods. To achieve a more robust convergence rate with respect to the possible large number of subdomains, two-level algorithms based on coarse space correction techniques are also studied in our numerical tests. Numerical results illustrate that two-level domain decomposition algorithms perform significantly better than the corresponding one-level algorithms when the number of subdomains becomes large. The numerical results (with timing) of parallel implementations of our proposed algorithms on massively parallel computers are currently undertaken and will be reported elsewhere. Our proposed *additive* Schwarz solvers and preconditioners, ASN and ASO, are anticipated to be highly parallelizable. Another future work is to estimate the condition number of our proposed algorithms as preconditioners to theoretically justify the observed very robust convergence performance of the preconditioned GMRES in our numerical results.

Upon completion of our current manuscript, we noticed some recent related work [16, 17], where some advanced 1-level optimized Schwarz methods based on certain parameterized Robin transmission conditions are proposed for obtaining better convergence rates, but their presented convergence analysis are technically more involved and the reported results are based on the first-order accurate backward Euler scheme in time and only limited to 1-level additive Schwarz nonoverlapping algorithm. Our current work provides more different domain decomposition algorithms (both 1-level and 2-level ones) with very comprehensive numerical comparison.

References

- [1] L. T. Biegler, M. Heinkenschloss, O. Ghattas, B. van Bloemen Waanders (Eds.), Large-Scale PDE-Constrained Optimization, Springer Berlin Heidelberg, 2003.
- [2] M. D. Gunzburger, Perspectives in flow control and optimization, SIAM, 2003.
- [3] M. Hinze, R. Pinnau, M. Ulbrich, S. Ulbrich, Optimization with PDE Constraints, Springer, 2008.
- [4] G. Biros, O. Ghattas, Parallel Lagrange–Newton–Krylov–Schur methods for PDE-constrained optimization. part I: The Krylov–Schur solver, SIAM Journal on Scientific Computing 27 (2) (2005) 687–713.
- [5] G. Biros, O. Ghattas, Parallel Lagrange–Newton–Krylov–Schur methods for PDE-constrained optimization. part II: The Lagrange–Newton solver and its application to optimal control of steady viscous flows, SIAM Journal on Scientific Computing 27 (2) (2005) 714–739.
- [6] E. E. Prudencio, R. Byrd, X.-C. Cai, Parallel full space SQP lagrange-newton-krylov-schwarz algorithms for PDE-constrained optimization problems, SIAM Journal on Scientific Computing 27 (4) (2006) 1305–1328.
- [7] A. T. Barker, X.-C. Cai, Two-level newton and hybrid schwarz preconditioners for fluid-structure interaction, SIAM Journal on Scientific Computing 32 (4) (2010) 2395–2417.
- [8] R. Chen, X.-C. Cai, Parallel one-shot Lagrange-Newton-Krylov-Schwarz algorithms for shape optimization of steady incompressible flows, SIAM J. Sci. Comput. 34 (5) (2012) B584–B605.
- [9] A. Borzì, V. Schulz, Computational optimization of systems governed by partial differential equations, SIAM, Philadelphia, PA, 2012.

- [10] O. Axelsson, S. Farouq, M. Neytcheva, Comparison of preconditioned krylov subspace iteration methods for pde-constrained optimization problems, *Numer. Algorithms* 73 (3) (2016) 631–663.
- [11] H. Yang, F.-N. Hwang, X.-C. Cai, Nonlinear preconditioning techniques for full-space lagrange-newton solution of PDE-constrained optimization problems, *SIAM Journal on Scientific Computing* 38 (5) (2016) A2756–A2778.
- [12] J. Nievergelt, Parallel methods for integrating ordinary differential equations, *Comm. ACM* 7 (1964) 731–733.
- [13] M. J. Gander, 50 years of time parallel time integration, in: T. Carraro, M. G. abd S. Körkel, R. Rannacher (Eds.), *Multiple Shooting and Time Domain Decomposition*, to appear.
- [14] M. Heinkenschloss, A time-domain decomposition iterative method for the solution of distributed linear quadratic optimal control problems, *J. Comput. Appl. Math.* 173 (1) (2005) 169–198.
- [15] A. T. Barker, M. Stoll, Domain decomposition in time for PDE-constrained optimization, *Comput. Phys. Commun.* 197 (2015) 136–143.
- [16] M. J. Gander, F. Kwok, On the time-domain decomposition of parabolic optimal control problems, in: *Proceedings of the 23rd International Conference on Domain Decomposition Methods*, Springer, 2016, pp. 207–216.
- [17] F. Kwok, Schwarz methods for the time-parallel solution of parabolic control problems, in: *Lecture Notes in Computational Science and Engineering*, 2016, to appear.
- [18] B. F. Smith, P. E. Bjorstad, W. D. Gropp, *Domain decomposition*, Cambridge University Press, Cambridge, 1996.
- [19] A. Toselli, O. Widlund, *Domain decomposition methods—algorithms and theory*, Springer-Verlag, Berlin, 2005.
- [20] T. P. A. Mathew, *Domain decomposition methods for the numerical solution of partial differential equations*, Springer-Verlag, Berlin, 2008.
- [21] E. E. Prudencio, R. Byrd, X.-C. Cai, Parallel full space SQP Lagrange-Newton-Krylov-Schwarz algorithms for PDE-constrained optimization problems, *SIAM J. Sci. Comput.* 27 (4) (2006) 1305–1328.
- [22] M. Heinkenschloss, M. Herty, A spatial domain decomposition method for parabolic optimal control problems, *Journal of Computational and Applied Mathematics* 201 (1) (2007) 88–111.
- [23] V. Dolean, P. Jolivet, F. Nataf, *An introduction to domain decomposition methods*, SIAM, Philadelphia, PA, 2015, algorithms, theory, and parallel implementation.
- [24] A. Quarteroni, A. Valli, *Domain Decomposition Methods for Partial Differential Equations*, Oxford University Press, 1999.
- [25] X. Deng, X.-C. Cai, J. Zou, Two-level space-time domain decomposition methods for three-dimensional unsteady inverse source problems, *Journal of Scientific Computing* 67 (3) (2015) 860–882.

- [26] M. J. Gander, L. Halpern, [Méthodes dedecomposition dedomaines notions de base](#), Encyclopédie électronique pour les ingénieurs.
URL <https://www.unige.ch/~gander/Preprints/Encyclopedia.pdf>
- [27] M. J. Gander, L. Halpern, K. S. Repiquet, A new coarse grid correction for ras/as, in: Domain Decomposition Methods in Science and Engineering XXI, Springer, 2014, pp. 275–283.
- [28] F. Kong, X.-C. Cai, A highly scalable multilevel schwarz method with boundary geometry preserving coarse spaces for 3d elasticity problems on domains with complex geometry, SIAM Journal on Scientific Computing 38 (2) (2016) C73–C95.
- [29] L. M. Carvalho, L. Giraud, G. Meurant, Local preconditioners for two-level non-overlapping domain decomposition methods, Numerical Linear Algebra with Applications 8 (4) (2001) 207–227.
- [30] L. Giraud, F. G. Vasquez, R. S. Tuminaro, Grid transfer operators for highly variable coefficient problems in two-level non-overlapping domain decomposition methods, Numerical Linear Algebra with Applications 10 (5-6) (2003) 467–484.
- [31] J.-L. Lions, Optimal control of systems governed by partial differential equations., Springer-Verlag, New York, 1971.
- [32] F. Tröltzsch, Optimal control of partial differential equations: theory, methods, and applications, AMS, 2010.
- [33] J. Liu, M. Xiao, A leapfrog semi-smooth Newton-multigrid method for semilinear parabolic optimal control problems, Comput. Optim. Appl. 63 (1) (2016) 69–95.
- [34] J. Liu, M. Xiao, A leapfrog multigrid algorithm for the optimal control of parabolic PDEs with robin boundary conditions, Journal of Computational and Applied Mathematics 307 (2016) 216–234.
- [35] R. LeVeque, Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems, SIAM, Philadelphia, PA, USA, 2007.
- [36] B. Jovanović, E. Süli, Analysis of Finite Difference Schemes: For Linear Partial Differential Equations with Generalized Solutions, Springer, 2013.
- [37] E. Efstathiou, M. J. Gander, Why restricted additive schwarz converges faster than additive schwarz, BIT Numerical Mathematics 43 (5) (2003) 945–959.
- [38] M. J. Gander, Schwarz methods over the course of time, ETNA. Electronic Transactions on Numerical Analysis 31 (2008) 228–255.
- [39] J. Xu, Iterative methods by space decomposition and subspace correction, SIAM Review 34 (4) (1992) 581–613.
- [40] J. Xu, J. Zou, Some nonoverlapping domain decomposition methods, SIAM Review 40 (4) (1998) 857–914.
- [41] E. E. Prudencio, X.-C. Cai, Parallel multilevel restricted schwarz preconditioners with pollution removing for PDE-constrained optimization, SIAM Journal on Scientific Computing 29 (3) (2007) 964–985.
- [42] E. Treister, I. Yavneh, Non-galerkin multigrid based on sparsified smoothed aggregation, SIAM Journal on Scientific Computing 37 (1) (2015) A30–A54.